

**AMENDMENTS TO THE DRAWINGS**

Applicant submits a replacement sheet for Figure 5 in which the misspelled term "course-grain" has been corrected to "coarse-grain." No new matter has been added.

### **REMARKS**

This amendment is responsive to the Office Action dated February 10, 2006. Applicant has made non-narrowing amendments to claims 5, 6, 25, 26, 39, 40 and 53. Claims 1-55 remain pending, with claims 16-18 and 36-55 currently being withdrawn due to restriction.

### **Claim Objections**

The Examiner pointed out that the term "coarse-grain" was misspelled. Applicant thanks the Examiner for identifying this error, and has amended claims 5, 6, 25, 26, 39, 40 and 53 to correct the misspelling of the term "coarse."

### **Restriction Under 35 U.S.C. § 121**

In the Office Action, the Examiner restricted claims 1-55 under 35 U.S.C. § 121 as follows:

- Group I.        Claims 1-15 and 19-35,
- Group II.       Claims 16-18 and 36-55.

During a telephonic conversation with the Examiner, Applicant provisionally elected Group 1 with traverse. Applicant affirms this election with traverse.

### **Claim Rejection Under 35 U.S.C. § 102**

In the Office Action, the Examiner rejected claims 1-3, 15, 22-24, and 35 under 35 U.S.C. 102(e) as being anticipated by Valois (USPN 2004/0260818). Applicant respectfully traverses the rejection.

Applicant submits that the present claims are easily distinguishable from the Valois reference. That is, Valois fails to disclose each and every feature of the claimed invention, as required by 35 U.S.C. 102(e), and provides no teaching that would have suggested the desirability of modification to include such features.

*Valois*

Valois describes a system and method for testing the security policies of a network device, and verifying the device implements its intended security policy.<sup>1</sup> In particular, Valois describes a software system that is a “*tool in diagnosing* the security vulnerability of a network device.”<sup>2</sup> FIG. 1 of Valois shows that the software tool verifies that a device or a plurality of devices correctly implements their intended security policies.<sup>3</sup> The software tool of Valois includes a configuration repository database 12, a security policy database 14, a test scripts database 16, a validation engine 18, and a parser engine 20. According to Valois, the test scripts database of the software testing system contains a collection of test scripts or expert rules that expresses a security characteristic or policy for testing the different network devices.<sup>4</sup> Valois describes that these test scripts of the verification software system may utilize regular expressions to search configuration files of the network devices to verify compliance with the desired security policies.<sup>5</sup>

In contrast to Valois, claim 1 is directed to a method comprising storing authorization data that defines an access control attribute and an associated regular expression specifying a textual pattern; evaluating a command using the regular expression to determine whether the command matches the textual pattern; and controlling access to configuration data of a device based on the evaluation. Claim 22 is substantially similar to claim 1 and is directed to a computer-readable medium. Applicant submits that the Valois software system for verifying network devices fails to teach or suggest nearly every one of these features.

First, Valois fails to teach or suggest storing authorization data that defines an access control attribute and an associated regular expression specifying a textual pattern. Claim 1 specifically requires that the authorization data itself define *both*: (1) an access control attribute, and (2) an associated regular expression. With respect to the requirement that the authorization data define an access control attribute, the Examiner cited Valois at 0058 and suggested that, in Valois, network devices store “authorization data” as configuration data having “references” to

---

<sup>1</sup> Valois at Summary

<sup>2</sup> Valois at 0068 (emphasis added).

<sup>3</sup> Valois at 0049.

<sup>4</sup> Valois at 0055.

<sup>5</sup> Valois at 0057, 0058.

access control lists.<sup>6</sup> The Applicant agrees with the Examiner in that access control lists indeed could be one form of an authorization attribute.

However, with respect to the claim requirement that the authorization data itself also define an associated regular expression, the Examiner then cited 0057 of Valois. As discussed above, 0057 of Valois refers to test scripts of an external verification software system that may utilize regular expressions to search configuration files of the network devices to verify compliance with the desired security policies.<sup>7</sup> Valois describes a test scripts database of the software testing system that contains a collection of test scripts or expert rules that expresses a security characteristic or policy for testing the different network devices.<sup>8</sup> By no means are these “test scripts” defined as part of the stored authorization data, as required by claim 1. To the contrast, Valois makes clear that the test scripts are stored within a database of a verification software system used to test multiple devices, which is entirely separate from the authorization data (access control lists) stored within those devices. Thus, Valois software verification system using test scripts to search configuration files on network devices for references to access control lists does not teach or suggest the requirement of claim 1 of “storing authorization data that defines an access control attribute and an associated regular expression specifying a textual pattern.” The plain language of claim 1 requires that the stored authorization data itself define *both*: (1) an access control attribute, and (2) an associated regular expression, and the access control lists of the network devices in Valois fail to store any form of a regular express specifying a textual pattern.

Second, Valois fails to teach or suggest evaluating a command using the regular expression to determine whether the command matches the textual pattern. Claim 1 specifically requires the evaluation of a *command* using the regular expression that was defined by the authorization data. With respect to these elements, the Examiner cited Valois at [0064], lines 1-5, and referred to use of a test program by the Valois software verification system to identify and assess the network devices use of an access control list. Here, and at [0065], Valois describes a plurality of test programs 26 that check a *configuration file* of a network device for improper usage of an access control list. Thus, Valois does not teach or suggest evaluation of *commands*

---

<sup>6</sup> See footnote 1 on pg. 5 of the Office Action.

<sup>7</sup> Valois at 0057, 0058.

<sup>8</sup> Valois at 0055.

using the regular expression at all. Valois describes parsing configuration files, not commands. Neither a configuration file nor a reference to an access control list within a configuration file is any form of a command. Moreover, Valois does not describe evaluation of a command using the regular expression *that was defined by the authorization data*, as required by claim 1. To the extent Valois uses regular expressions, they are in the form of an external test program (script) 26 to test references to authorization data (ACLs), and are not defined by stored authorization data, as required by claim 1, and used to evaluate commands, as further required by claim 1.

Third, Valois does not teach or suggest controlling access to configuration data of a device based on the evaluation of the command, as required by claim 1. With respect to these elements, the Examiner cited Valois at [0066], lines 1-9, and referred to the validation engine of the Valois testing and verification software system. Applicant disagrees that the validation engine of the Valois testing and verification software system in any way actually *controls* access to configuration data of a device based on the evaluation of the command, as required by claim 1. Valois makes clear that the system is merely a “tool in *diagnosing* the security vulnerability of a network device.”<sup>9</sup> As a tool, the validation software system reports that a device either “passes” result or a “fails” the test and, if it fails, the software system lists the access control lists improperly referenced by a given device.<sup>10</sup> The Valois tool provides no mechanism by which a regular expression can be used to actually control access to configuration data of a device. Thus, Valois provides no teaching or suggestion actually *controlling access to configuration data* of a device based on the evaluation of a command using a regular expression defined by authorization data.

With respect to dependent claims 2-3, Valois fails to anticipate the claim requirements for the reasons set forth above. Claim 2, for example, specifically requires that controlling access comprise allowing access to the configuration data when the textual pattern of the regular expression matches the command. Claim 3 requires that controlling access comprises denying access to the configuration data when the textual pattern of the regular expression matches the command. With respect to these elements, the Examiner refers to the pass/fail output from the Valois test program 26 as described in 0067. As discussed above, Valois provides no teaching or

---

<sup>9</sup> Valois at 0068 (emphasis added).

<sup>10</sup> Valois at 0067.

suggestion of actually *controlling access to configuration data* of a device, let alone controlling access based on the evaluation of a command using a regular expression defined by authorization data. Moreover, the Valois verification tool outputs “pass” result if a *match* is detected by the test program. Applicant requests clarification as to how the “pass” result from the Valois verification tool could teach or suggest *denying* access to the configuration data when the textual pattern of the regular expression *matches* the command, as required by claim 3.

In order to support an anticipation rejection under 35 U.S.C. 102(e), it is well established that a prior art reference must disclose each and every element of a claim. This well known rule of law is commonly referred to as the “all-elements rule.”<sup>11</sup> If a prior art reference fails to disclose any element of a claim, then rejection under 35 U.S.C. 102(e) is improper.<sup>12</sup> Valois fails to disclose each and every limitation set forth in claims 1-15, and provides no teaching or suggestion of modification to include such features. For at least these reasons, the Examiner has failed to establish a prima facie case for anticipation of Applicant’s claims 1-3, 15, 22-24, and 35 under 35 U.S.C. 102(e). Withdrawal of this rejection is requested.

### **Claim Rejection Under 35 U.S.C. § 103**

#### *Claim 4*

In the Office Action, the Examiner rejected claim 4 under 35 U.S.C. 103(a) as being unpatentable over Valois (USPN 2004-0260818) as applied to claims 1-3, 15, 22-24 and 35 above, and further in view of Mitra (USPN 6,973,460). Applicant respectfully traverses the rejection. Claim 4 requires that storing authorization data comprises storing the authorization data as an authorization class that conforms to a class syntax. That is, in view of the elements of claim 1, claim 4 requires storing authorization data as an authorization class that conforms to a class syntax, and that the authorization data define *both*: (1) an access control attribute, and (2) an associated regular expression.

---

<sup>11</sup> See *Hybritech Inc. v. Monoclonal Antibodies, Inc.*, 802 F.2d 1367, 231 USPQ 81 (CAFC 1986) (“it is axiomatic that for prior art to anticipate under 102 it has to meet every element of the claimed invention”).

<sup>12</sup> *Id.* See also *Lewmar Marine, Inc. v. Barient, Inc.* 827 F.2d 744, 3 USPQ2d 1766 (CAFC 1987); *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (CAFC 1990); *C.R. Bard, Inc. v. MP Systems, Inc.*, 157 F.3d 1340, 48 USPQ2d 1225 (CAFC 1998); *Oney v. Ratliff*, 182 F.3d 893, 51 USPQ2d 1697 (CAFC 1999); *Apple Computer, Inc. v. Articulate Systems, Inc.*, 234 F.3d 14, 57 USPQ2d 1057 (CAFC 2000).

With respect to these elements, the Examiner cites Mitra, col. 8, ll. 7-18, and states that it would be obvious to one of ordinary skill to modify the Valois verification tool in view of the teachings of Mitra so that the “authorization data” of Valois was stored as a class. The Applicant points out that the Examiner is suggesting that Valois somehow be modified to use a class to store *both* the access control lists (which are located within the network devices being tested) and the regular expression testing programs / scripts of the Valois verification tool. The Examiner has pointed to no evidence in any of the references of record, either singularly or in combination, as to technically how or for what reason a single class could be used to store this disparate information. As discussed above, the test scripts of the Valois verification tool are fundamentally NOT authorization data, and the use of regular expressions by those test scripts is unrelated to the access control lists stored by the network devices in order for authentication purposes. Fundamentally, the network devices ONLY use the access control for security purposes, and suggesting that a class be used to store regular expressions from the Valois test scripts goes well beyond any teaching or motivation from the references.

*Claims 5-11 and 25-31*

In the Office Action, the Examiner rejected claims 5-11 and 25-31 under 35 U.S.C. 103(a) as being unpatentable over Valois (USPN 2004-0260818) and in further view of Delany (USPN 2002/0156879). Applicant respectfully traverses the rejection.

With respect to claim 5, the Examiner correctly recognizes that Valois fails to teach or suggest storing authorization data that includes a coarse-grain access control attribute defining access control rights for respective groups of resources provided by the device, and controlling access comprises controlling access to the configuration data based on the coarse-grain access control attribute and the evaluation of the regular expression. However, the Examiner cites Delany at [0118] and appears to suggest that it would have been obvious to one of ordinary skill in the art to modify the Valois verification tool in view of the teachings of Delany so that the access control lists of (“authorization data”) of the network devices of Valois use host names and URL prefixes as taught by Delany as a coarse-grain attribute. Then, with respect to the remaining elements of claim 5, the Examiner suggests that it would have been obvious to one of ordinary skill to somehow modify the Valois testing tool to control access to the configuration

data using both the host names and URL prefixes of the modified access control lists as well as the evaluation of the regular expression. Applicant submits that even if the access control lists of Valois were modified as suggested by the Examiner, there is still no teaching or suggestion in the combination of references for controlling access using authorization data that defines both a coarse-grain access control attribute as well as a regular expression for evaluation of a command.

With respect to claim 7, paragraph 0199 of Delany describes a *graphical* user interface (GUI), which is certain not a command-line interface, which is textual and non-graphical. Moreover, Delany at 0199 provides no suggestion of entering a text-based command at all.

Claim 8 requires evaluating the command in real-time while the client inputs the command via the command line interface. Thus, the literal language of claim 9 requires evaluation using the regular expression (per claim 1) in real-time while the client inputs the command via a command line interface. In rejecting claim 9, the Examiner appears to suggest that it would be obvious to modify the Valois testing tool to perform such functions based on the teaching of Delany at [0199]. This assertion, however, is incorrect for several reasons. First, Delany at 0199 refers to a *system administrator* that employs an Online Certificate Status Protocol ("OCSP") to check the status of a *certificate* revocation in real time through an online connection with Certificate Authority 2084. In other words, Delany states that a system administrator can check the current (i.e., "real-time") status of whether a digital certificate has been revoked. Thus, Delany has nothing whatsoever to do with evaluation of commands at all, and provides no teaching or suggestion as to how commands could be evaluated in real-time while the commands are being entered using a command line interface. Checking a status of a digital certificate is completely and entirely unrelated to evaluation of commands, and provides no teaching as to how commands could be evaluate in real-time as the commands are received via a command line interface. Second, modification of the Valois testing tool to check whether digital certificates have been recalled still provides no teaching or suggestion of evaluation of a command in real-time. The Valois test programs 26 searches for and parses configuration files stored on network devices. How could this in any way be applied in real-time as a command received from a client, even in view of Delany's discussion of checking the status of digital certificates? Certainly, the configuration files on the network devices are not "commands" received from clients. Moreover, how could a test tool that parses configuration files and



identifies improper references be applied in real-time at all while a command is received from a client?

Claim 9 requires that the configuration data is arranged in the form of a multi-level configuration hierarchy having a plurality of objects, and each of the objects represents a portion of the configuration data that relates to one or more resources of the device. In rejecting claim 9, the Examiner refers to Delany at [0142] that describes a directory tree that describe profiles for users and groups within an organization. This certainly is not "configuration data" for a network device, and Valois in view of Delany fails to teach or suggest any of the requirements of claim 9.

With respect to claims 10-11, the Examiner appears to suggest that, in view of the Delany directory tree of user profiles, one of ordinary skill in the art would modify the Valois testing tool, which looks for improper ACL references, to use regular expressions that define the textual pattern to match the textual labels of configuration data. This fails to recognize at least that: (1) Delany is not describing configuration data, and (2) one would not modify the regular expression of the Valois testing tool to match labels for configuration data as this would defeat the stated purpose of the Valois testing tool, which is to identify improper reference to authorization data (ACLs).

*Claims 12-14, 19-21, and 32-34*

In the Office Action, the Examiner rejected claims 12-14, 19-21, and 32-34 under 35 U.S.C. 103(a) as being unpatentable over Valois (USPN 2004-0260818) in view of Delany (USPN 2002/0156879) and further in view of Nelson (USPN 6,243,713). Applicant respectfully traverses the rejection. The applied references fail to disclose or suggest the inventions defined by Applicant's claims, and provide no teaching that would have suggested the desirability of modification to arrive at the claimed invention.

With respect to many of these claims, the Examiner appears to have confused pre-processing a regular expression itself, as required by Applicant's claims, with pre-processing of text within a document. As explained in the present application, a regular expression may be pre-processed (i.e., before evaluation of a command) to automatically insert meta-characters to ensure the regular expressions are correctly formulated in view of the hierarchical arrangement of configuration data. Pre-processing the regular expression may allow the regular expression to

subsequently be applied in real-time to evaluate a command as a user enters that command. See, e.g., [0044]-[0045] of the present application.

Claim 12, for example, requires pre-processing the regular expression to automatically insert one or more meta-characters into the regular expression based on the hierarchical arrangement of the configuration data. Claim 13 requires receiving the command from a client via a command line interface; and pre-processing the regular expression so that the command is evaluated with the regular expression in real-time as the client enters the command. Claim 14 requires evaluating the command with the pre-processed regular expression each time the client enters a token indicating a textual break within the command.

In contrast, Nelson, col. 10, ln.39-50, describes pre-processing text within a document, such as an RTF document, and not pre-processing a regular expression that is used to evaluate text entered by a user. There is no suggestion of pre-processing a regular expression, as required by claim 12.

Moreover, the combination of references asserted by the Examiner provides no suggestion of using the regular expression to evaluate commands in real-time as a client enters the command, as required by claim 13. The Examiner's rejection of claim 13 is erroneous for at least the following reasons: (1) Delany at [0199] describes a *Graphical* User Interface and, contrary to the Examiner's position, provides no suggestion of any form of textual command, (2) Nelson describes pre-preprocessing text within a document and does not suggest pre-processing a regular expression, and (3) the Examiner's reliance on Delany's use of a protocol to determine the current status of a digital certificate provides no teaching or suggestion of actually evaluating commands in real-time.

With respect to claim 14, the Examiner cites Nelson at column 17, lines. 35-40. However, at this passage, Nelson merely states that a user may supply a regular expression, which is then applied to a list of tokens generated during the multimedia retrieval process of Nelson. Importantly, in Nelson, the user has entered the regular expression. That is, the regular expression is the user input, and that regular expression is being used as a pattern to build multimedia queries. Quite the opposite, claim 14 requires evaluating a command with the pre-processed regular expression each time the client enters a token indicating a textual break within the command. Neither Nelson nor any of the other references describes using regular expressions

to evaluate commands entered by the user, let alone in real-time as the command is being entered.

For at least these reasons, the Examiner has failed to establish a prima facie case for non-patentability of Applicant's claims under 35 U.S.C. 103(a). Withdrawal of this rejection is requested.

### CONCLUSION

All claims in this application are in condition for allowance. Applicant respectfully requests reconsideration and prompt allowance of all pending claims. Please charge any additional fees or credit any overpayment to deposit account number 50-1778. The Examiner is invited to telephone the below-signed attorney to discuss this application.

Date:

By:

May 10, 2006  
SHUMAKER & SIEFFERT, P.A.  
8425 Seasons Parkway, Suite 105  
St. Paul, Minnesota 55125  
Telephone: 651.735.1100  
Facsimile: 651.735.1102

Kent J. Sieffert  
Name: Kent J. Sieffert  
Reg. No.: 41,312